Representation and execution of temporal criteria for guideline-based medical decision support at the Intensive Care Unit

Triantafyllou, Evangelia; Kokkinou, Eleni; de Clercq, Paul; Peek, Niels; Korsten, Hendrikus H. M. ; Hasman, Arie

Published in:
Proceedings of the 17th Belgium-Netherlands conference on Artificial Intelligence (BNAIC)

Publication date:
2005

Document Version
Accepted author manuscript, peer reviewed version

REPRESENTATION AND EXECUTION OF TEMPORAL CRITERIA FOR GUIDELINE-BASED MEDICAL DECISION SUPPORT AT THE INTENSIVE CARE UNIT

Evangelia Triantafyllou [a] Eleni Kokkinou [b] Paul A. de Clercq [c]

Niels Peek [a] Hendrikus H. M. Korsten [b] Arie Hasman [a]

[a] *Department of Medical Informatics, Academic Medical Center, University of Amsterdam, Amsterdam, The Netherlands*

[b] *Division of Signal Processing Systems, Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands*

[c] *Medecs BV, Eindhoven, The Netherlands*

Abstract

This paper presents the development of a framework for representation and execution of temporal criteria in guideline-based Medical Decision Support Systems (MDSSs). An MDSS is a system that provides medical advice based on patient information and on knowledge of medical best practices as represented in ontologies or other forms of knowledge representation. The proposed framework consists of a temporal ontology that defines time-related concepts and their interrelationships, a temporal data model, a mechanism for executing temporal criteria, and the interfaces with data sources. In order to extract requirements for our framework, we studied practice guidelines used at the ICU and we contacted health-care professionals. We implemented our framework in the Gaston MDSS and connected it to the database of the ICU department of the Catharina Hospital in Eindhoven, the Netherlands. We tested it and found it to provide a solid base for temporal functionality in a MDSS. We will further evaluate it in a clinical environment.

1 Introduction

Medical decision support systems (MDSSs) are computer systems that permit explicit representation of medical knowledge and use both clinical data and this knowledge to help health-care professionals make clinical decisions. A common type of MDSSs is the so-called guideline-based system. Such a system collects data from the patient record and/or from patient monitoring systems. These data are needed to apply the medical knowledge captured in clinical practice guidelines. In order to be accessible by the system, this knowledge should be represented in a computer interpretable format. Ontologies are forms of knowledge representation, which describe a domain in terms of concepts, their attributes, and the relationships that exist between these concepts [6]. After consulting the actual patient data stored in information systems or databases, and the practice guidelines, the system provides the clinical staff with medical advice. The medical advice may consist of warnings, alarms, reminders, etc.

Time plays a very important role in medicine. Disease patterns unfold in time and patient care involves a cycle of observation, decision making, and possibly treatment and intervention. Therefore, the integration of time information into guideline-based MDSSs is critical for improving the quality of their support.

We will refer to the decision criteria in practice guidelines that are based on time-varying patient data as *temporal criteria*. Various approaches exist on how to represent this kind of decision criteria and execute them in a MDSS. For the representation of temporal criteria a representation language is needed. The execution of temporal criteria involves querying the information system that is coupled to the medical decision support system.

Asbru [12] and EON [14] support temporal knowledge representation and execution of temporal criteria during guideline execution. Asbru is a rich guideline representation language regarding temporal aspects but approaches to make it fully executable are still in progress. Currently there is an interpreter for Asbru, called Asbru Light+ [10], which covers a subset of the temporal aspects of Asbru (namely, the handling of temporal uncertainty and real time constraints). EON supports temporal criteria during execution, but the representation language is limited with respect to the temporal information it can represent. For instance, there is no support for absolute time intervals (intervals defined by a start and an end time instant).

This paper describes a new framework to represent and evaluate temporal criteria in clinical practice guidelines. The framework aims to support more temporal expressiveness than is available in state-of-the-art execution engines like Asbru Light+ and EON, and yet be fully executable and ready to be used in clinical practice. For this reason, we have chosen to support only temporal criteria that are frequently encountered in clinical guidelines.

The proposed framework contains a) a *temporal ontology*, which defines time-related concepts (e.g. time points or intervals) and their interrelationships, namely temporal operators (before, after, during, etc) that relate these concepts and are needed in order to reason about time, b) a *temporal data model*, and c) a mechanism for executing temporal criteria (*temporal query mechanism*). In order to design this framework we integrated methods used in other approaches aiming at creating a working system that can be used in clinical practice.

The developed framework was implemented in the Gaston environment for MDSS development, which was built at Eindhoven University of Technology and is now a commercial product of Medecs company [7]. Interfaces with data sources (e.g. databases) and compatibility with the knowledge representation of Gaston were sought. By implementing the framework in a MDSS we were able to test it in clinical practice. The developed framework fulfils the requirements regarding representation and execution of the selected set of temporal criteria.

2 Temporal criteria in Gaston

Gaston is a rapid development environment for MDSSs that is based on domain and method ontologies. Some of the systems developed in Gaston have already been implemented in hospitals and are being used in daily practice [2, 4, 5].

After studying various clinical practice guidelines used at the Intensive Care Unit (ICU) department of the Catharina Hospital and after consulting its clinical staff, we extracted the following categories of temporal criteria.

- Criteria regarding values of time-oriented data. These criteria are not temporal regarding representation aspects. But they are temporal on the database level of hospital information systems because the laboratory tests and the medications are always accompanied by a timestamp, and are stored in temporal tables in such databases. Examples of such criteria are:
  - Arterial temperature[†] was greater than 38℃.
  - The dose of Nutrison was greater than 70 mg.
  - Diuretics were given to the patient.
- Criteria regarding the temporal dimension of time-oriented data. For this kind of criteria, temporal operators are necessary. These operators may relate a parameter to a point in time or to another parameter. When reasoning about time, abstract concepts are usually necessary, such as today, last, etc. Examples or such criteria are:
  - Arterial temperature was measured before today.
  - Loperamide was prescribed less than 2 days after the last dose of Nutrison.
- Criteria regarding both values and temporal dimension of time-oriented data, for example:
  - The dose of Nutrison was greater than 40 mg and was prescribed less than 3 days before 13/05/2005 15:00:00.

[†]Arterial temperature is the same temperature as the blood flowing from the heart.

In our opinion, these criteria are representative of the criteria found in practice guidelines used at the ICU and they contain enough complexity concerning representation and execution aspects. Gaston should be able to represent and execute criteria belonging to any of the three categories in order to support decision making at the ICU.

The version of Gaston without the framework for representation and execution was able to execute simple temporal criteria without incorporating the notion of time. That was possible by handling time values as numerical values. Therefore, it could execute criteria belonging to the first category and some of the criteria belonging to the second category. However, it could not support at all temporal criteria regarding both values and temporal dimension of temporal data (third category). Moreover, it could only use the actual moment ("now") as a reference for temporal reasoning (e.g. more/less than two days ago, during the last five hours). No temporal operators between time instants or intervals were supported.

Our framework is able to represent and execute in Gaston criteria belonging to all three aforementioned categories and therefore enhanced the performance of Gaston regarding representation and execution of temporal information.

3 The temporal ontology

3.1 A metamodeling approach

For developing the temporal ontology of the framework, we used a metamodel as a collection of concepts that describe the time domain. The metamodel helped us to find time-related concepts and to adjust them to the needs of temporal knowledge representation. The UML (Unified Modelling Language) Profile for Schedulability, Performance, and Time Specification was used to provide these concepts [13].

This specification consists of a set of interpretation rules, markings, and constraints imposed on standard UML to more accurately capture the specific phenomena of a given application domain. Not all the concepts defined in this profile are relevant to our application domain. We used only concepts from the second sub-profile of this core framework, which contains a very general model of physical and measured time (RTtimeModeling).

We first represented the ontology as a model in UML and then we imported it to an ontology development environment, namely Protégé [8], that offers the possibility to import UML schemas directly in the ontology editor.

3.2 Description of the temporal ontology

In the ontology there is a distinction between concrete and abstract concepts. Concrete concepts are the ones used directly by the modeler, whereas abstract concepts are used to define common features of two or more related concepts. The developed temporal ontology is shown by the UML diagram in Figure 1 in the form of a conceptual model.

The temporal ontology considers physical time as a continuous and unbounded progression of physical time instants, as perceived by some observer, such that it is a fully ordered set. Time in the temporal ontology is considered to be discrete, which means that time is broken up into quanta. Furthermore it is assumed that physical time progresses monotonically and only in the forward direction.

The basic time-related concepts contained in the ontology are the following:

1. PhysicalTime: This is an abstract concept that represents the notion of physical time. It is represented indirectly through time values of physical instants or periods.

2. PhysicalInstant: This is the abstract concept of a physical time instant – a point in time. Like geometrical points, instants do not have any extent (duration). Physical instants are represented through measurements, which are time values.

3. PhysicalPeriod: This is the abstract concept of a physical period of time. Physical periods are represented through measurements, which are durations.

4. Duration: This represents the elapsed time between two time instants. Durations are represented directly by their attributes "period" that is a number, and "unit" that is a string. The unit declares the time unit of the period (e.g. 2 hours, 35 min).

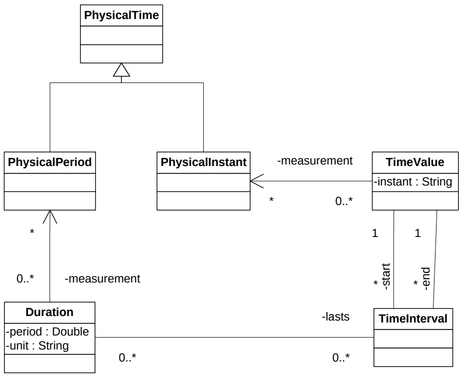

Figure 1: Basic time modeling concepts. Physical time can be either instants or periods. There is direct support for representing time, namely through measurements. Each TimeValue can be defined by one or more measurements depending on the existence or not of different frames of reference. TimeIntervals can be represented either by their start and end TimeValue or as durations.

5. TimeValue: A value that corresponds to a particular physical instant in time as measured in some frame of reference. The time values are represented directly through their attribute "instant", which is a string. This string can be an absolute time (e.g. 04/09/2004 18:00:00) or a temporal abstraction (e.g. now).

6. TimeInterval: A definite length of time. Time intervals start and end at specific points in time and they have a duration.

7. Temporal operators: The temporal operators are also defined as possible relations between physical instants and/or intervals. The ontology supports a subset of the Allen's operators [1]. More specifically the temporal operators BEFORE, AFTER, EQUALS, DURING, and OVERLAP are included. The rest of the operators defined by Allen are not supported because they were not necessary for the representation needs of the temporal criteria we chose. The definition of temporal operators allows for uncertainty in the definition of the time values or intervals related with the operators. More specifically, there are two parameters connected with each operator, min and max, and they are used to shift the reference time value to the past or the future, respectively.

In the developed temporal ontology some abstract concepts of the profile had to become concrete or to be defined in a different way (e.g. the definition of duration). New concepts were also included (e.g. PhysicalPeriod). Moreover, some temporal operators that were not modeled in this profile were added. These additions were necessary in order to cover the different types of temporal criteria (section 2).

To give an example, a criterion in a guideline may state whether arterial temperature was measured before today. In Figure 2, the representation of this example with instances of classes of the temporal ontology is given. The notation in Figure 2 is not a standard UML notation and shows instances of classes needed to represent the temporal criterion, and their interrelationships. The class of Arterial Temperature belongs to the medical domain ontology. Note that today is assumed to be a time value (and not a time interval), presented as a date.

4 Temporal query system

4.1 A mediator architecture

In medical decision support, retrieving time-related information from patient data sources is called a temporal query. A system that is able to perform and evaluate such queries is called a temporal query system. The problem of querying databases with respect to time concerns the non-explicitness of time information in the databases. Time information is usually just a field, whose name and data type are defined by the database designer and may vary between different databases. In order for the MDSS (or any other application) to use time information, it would have to pose different

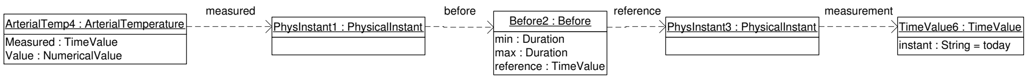

Figure 2: Representation of the temporal criterion "Arterial temperature measured before today?". The dashed arrows show relationships between instances. Each such arrow points to an object that is connected with an attribute of another object. The arrows are named after the attributes.

temporal queries for each database and execute them in a different way. However, a decision support system needs to be decoupled from the data sources, to maximize reusability and modularity.

Also, time information (e.g. valid time, time of data entry) contained in the data sources should be made explicit. If the time information is not formally stated, then operations on the temporal data have the risk of being meaningless and conceptually wrong. The MDSS should communicate with the temporal query system assuming a standard formal structure for the temporal data.

For these reasons we chose a *mediator architecture* for the temporal query system of the framework. A mediator is an intermediate layer of processing between the application (MDSS) and the database. It is used to decouple the database from the application. The application only communicates with the mediating system and only has one view of the data. The mediator query system has the capability to connect with different databases. In addition, the mediating system presents a formal temporal data model to the MDSS. It acts both as a connection between the MDSS and the data-sources and as a layer that adds temporal structure to the data and temporal query capability. To perform this task, the mediating temporal query system should have a temporal data model and also well defined interfaces with the MDSS and the databases.

4.2 Description of the temporal query system

We propose an overall architecture for the temporal query system (Figure 3), which aims to facilitate temporal queries to the underlying medical database and consists of: a) a temporal data model for the MDSS, b) a set of operators and their execution mechanisms that enable the MDSS to perform temporal queries, c) an interface between external databases and the temporal data model of the MDSS, d) a mapping module between the domain concepts used by the MDSS in the query and the domain concepts used in the temporal data model. In the following sections we will describe how the temporal model is built on top of an underlying data source.

4.2.1 Temporal data model

In order to support temporal questions, the query system has to support an internal data representation that includes time-related information. We propose a data representation which we will call "internal temporal model". It is essentially a temporal relational database, which acts as a temporal extension of the medical databases (data-sources) that the MDSS uses. A temporal data model defines how time and its attributes are represented, how facts are associated with time and the language that is used to apply operations on these attributes.

The non-temporal data types in the model will be represented by standard data types like float, integer, char etc. User-defined time-related data, such as date of birth will be represented by datatypes like date, time or date-time. The time-related data that have specified meaning in the model, will always be represented by a timestamp. A timestamp contains both date and time. Our temporal database uses valid-time [9]. In a valid-time table, one column always contains a time-stamp that indicates the validity of each row valid at that time-instant. We call this type of table an *event table*. The other type of valid-time table is the *state table*. The state table has two timestamp columns, indicating the time span that each row is valid. Finally, there are also non-temporal tables where no special meaning is given to time. We use event tables to represent instants, and state tables to represent periods. The notation is similar to the one used by O'Connor [11].

After defining a temporal database with a valid time-model, we made sure that it follows all integrity constraints of a database. Moreover, when we added temporal fields (the valid-time columns), we included them in the primary key in order for it to be unique. Finally, we preserved the time-related meaning that each row has. The temporal database is created when a guideline is

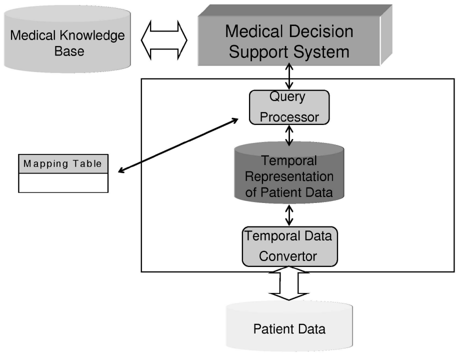

Figure 3: The proposed architecture for the temporal query system. The MDSS uses medical knowledge that includes temporal criteria. It also uses patient data. The patient data are fitted into a temporal representation with the help of a program called temporal data convertor. The evaluation of the temporal criteria against the patient data is done by the query processor.

executed on data from a given patient, since each guideline is executed for one patient. These data are updated with each new guideline execution for the specific patient.

4.2.2 Set of queries

We defined a set of temporal operators that can be applied to this temporal data model. Different temporal operators are applied to state and event tables. In state tables, temporal operators apply to intervals while in event tables they apply to specific time instants. A list of the supported temporal operators can be found in section 3.2.

4.2.3 Query mechanism

The language used to access the data in the temporal database is SQL. The temporal operators described in section 3.2 are implemented as parameterized queries written in SQL. A query mechanism, meaning an application layer above the temporal database receives the query from the MDSS, extracts the type of each operator and the needed parameters to execute the query, and creates the correct SQL query which is executed on temporal databases.

4.3 Example of execution of a temporal criterion

Let us see how the whole procedure works on a simple example temporal query “Was Arterial temperature measured before today?”. The execution engine is responsible for parsing these criteria. When temporal criteria must be executed, the execution engine calls the query module and sends each criterion in a format the query mechanism understands. From this format, the query module extracts information about:

- Domain concepts, such as Arterial Temperature and specifically
  - Name of the temporal table containing the domain concept (in this example “Laboratory”)
  - Name of the field of the temporal table mapped to the domain concept attribute (in this example “LabTest”)
  - Type of table, which may be state, event or non-temporal (in this example event)
- Temporal operators (in this example, “before”)
- Comparison value used by temporal operators (in this example “today” that is translated to a date, “15/06/2005”)
- Non-temporal operators (in this example none)
- Comparison values of non-temporal operators (in this example none)

The names of the temporal fields of the temporal database are always fixed. The query mechanism gets these names from a mapping module.

The query mechanism then provides parameters for the query. The parameterized SQL code for the event operator BEFORE is:

```
SELECT TableName.FieldName,TableName.TemporalFieldName
FROM TableName
WHERE TableName.FieldName Non-TemporalOperator Non-TemporalValue
AND TableName.TemporalFieldName < TemporalValue
```

For the above example the produced query is:

```
SELECT Laboratory.LabTest, Laboratory.IssueDate
FROM Laboratory
WHERE Laboratory.LabTest="Arterial Temperature"
AND Laboratory.IssueDate <  "15/06/2005";
```

The SQL query is executed over the temporal relational database that contains the structured patient data. The result is a set of tuples that satisfy the query, that is sent back to the execution engine of the MDSS.

The above-mentioned example is relatively simple. However, the temporal query mechanism is more sophisticated. It is designed to support not only simple temporal criteria such as the example, but also more complex ones that combine two domain concepts.

5 Integration with Gaston

In order to add the framework to Gaston, we made the following changes in Gaston. Firstly, the temporal ontology with all the aforementioned classes and relationships was added to the Gaston ontologies using Protégé. Secondly, the intermediate representation language that represents the guidelines as composed in the guideline editor of Gaston, was altered in order to support representations using the new temporal ontology. The guideline execution engine of Gaston was also tuned with the new version of the representation language. Finally, the temporal query mechanism was added to Gaston and was connected to the ICU database of the Catharina Hospital in Eindhoven for testing. This procedure should be followed every time that the framework is connected to a MDSS.

By connecting our framework to an ICU database, we were able to test its functionality and performance. We found that the developed framework supports representation and execution of all the aforementioned categories of temporal criteria. The execution time for this kind of criteria varied from a few milliseconds to several seconds (in any case less than five), depending on the size of the database tables involved.

6 Discussion

To sum up, this work provides a solid basis for providing temporal functionality to a guideline based MDSS used at ICU departments. It deals with representation and execution of temporal criteria, integration of data-sources, and with time in medical databases. The framework proposed in this work was conceived and created as an independent tool and therefore can be extended and reused by other applications that need to represent and query time-related information. The final system was tested in Gaston and was found functional, therefore contributing in practice to the improvement of health-care services, by providing new and better medical decision support.

Although many groups have developed guideline representation languages supporting temporal criteria, this is one of the few cases where the guideline representation language can be also fully executed. The adopted representation is not very rich, like Asbru, but expressive enough for the temporal decision criteria commonly found in clinical practice guidelines. The representation language is more expressive than EON, which also supports execution of temporal criteria.

Uncertainty is also a fundamental aspect of temporal reasoning in the clinical domain and the ontology provides for it. Firstly, multiple frames of reference are allowed in time definition. This means that one time instant may be connected with multiple time measurements depending on the frame of reference. Secondly, uncertainty is supported in combination with temporal operators. The parameters min and max of such operators allow for uncertainty when reasoning about time.

The resulting temporal query system is in many cases similar to the Cronus II system [11] and to IDAN [3]. It uses the mediator approach and provides a temporal model on top of the databases, similarly to these systems. The temporal model is also relational and supports valid-time as the model in these systems. Nevertheless, Cronus II and IDAN utilize a temporal query language, which is an extension of SQL while this approach supports a set of predefined SQL queries that implement the requested temporal operators. We believe that our implementation of a temporal query system is different from the existing ones because it is simple, practical, and it focuses on the functionality needed by the decision support system.

In the future, we will use the developed framework in Gaston to create a system, which will guard the logistics and medical progress of the stem cell transplantation (SCT) procedure at the Haematology department of the Academic Medical Centre in Amsterdam, The Netherlands. The SCT involves different type temporal criteria, such as criteria regarding synchronization and workflow aspects. Therefore, we expect to enhance our framework by expanding its temporal knowledge representation and execution performance.

References

[1] J. F. Allen. Towards a General Theory of Time and Action. *Artif Intell*, 23:123–154, 1984.

[2] R. Bindels, P. A. de Clercq, R. A. G. Winkens, and A. Hasman. A test ordering system with automated reminders for primary care based on practice guidelines. *Int J Med Inf*, 58-59(1):219–233, 2000.

[3] D. Boaz and Y. Shahar. IDAN: A distributed temporal-abstraction mediator for medical databases. In *Proc. AIME*, pages 21–30, Cyprus, 2003.

[4] P. A. de Clercq, J. A. Blom, A. Hasman, and H. H. M. Korsten. A strategy for development of practice guidelines for the icu using automated knowledge acquisition techniques. *Int J Clin Monit Comput*, 15:109–117, 1999.

[5] P. A. de Clercq and A. Hasman. Design of a consumer health record for supporting the patient-centered management of chronic diseases. In *Proc. MedInfo*, pages 1445–1449, UK, 2001.

[6] P. A. de Clercq, A. Hasman, J. A. Blom, and H. H. M. Korsten. The application of ontologies and problem-solving methods for the development of shareable guidelines. *Artif Intell Med*, 22(1):1–22, 2001.

[7] P. A. de Clercq, A. Hasman, J. A. Blom, and H. H. M. Korsten. Design and implementation of a framework to support the development of clinical guidelines. *Int J Med Inf*, 64(2–3):285–318, 2001.

[8] J. Gennari et al. The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. Report SMI-2002-0943, Stanford University, 2002.

[9] R.T. Snodgrass et al. *TSQL2 Temporal query language*. Kluwer Academic Publishers, Boston, 1995.

[10] C. Fuchsberger and S. Miksch. Asbru's Execution Engine: Utilizing Guidelines for Artificial Ventilation of Newborn Infants. In *IDAMAP & KBIM-AIC*, Cyprus, 2003.

[11] M. J. O'Connor, S. W. Tu, and M. A. Musen. The Chronus II temporal database mediator. In *Proc. AMIA Symp*, pages 567–571, 2002.

[12] Y. Shahar, S. Miksch, and P. Johnson. The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artif Intell Med*, 14(1–2):29–51, 1998.

[13] The Object Management Group and (OMG). UML$^{\text{TM}}$ Profile for Schedulability, Performance, and Time Specification. Adopted Specification ptc/02-03-02, 2002. http://www.omg.org.

[14] S. W. Tu and M. A. Musen. Modeling Data and Knowledge in the EON Guideline Architecture. In *Proc. MedInfo*, pages 280–284, UK, 2001.